

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants: RAMAMOORTHY, et al. Patent Application  
Application No.: 10/690,605 Group Art Unit: 2436  
Filed: October 23, 2003 Examiner: Hoang, Daniel L.  
For: SMART TRANSLATION OF GENERIC CONFIGURATIONS

REPLY BRIEF

In response to the Examiner's Answer mailed on December 21, 2011, Appellants respectfully submit the following remarks.

## REMARKS

Appellants are submitting the following remarks in response to the Examiner's Answer. In these remarks, Appellants are addressing certain arguments presented in the Examiner's Answer. While only certain arguments are addressed in this Reply Brief, this should not be construed that Appellants agree with the other arguments presented in the Examiner's Answer.

### Examiner's Answer page 8 lines 16-20

The Examiner's Answer states on page 8 lines 16-20, "Whipple does indeed teach translating a native format into another native format but Whipple further teaches that the native format is first translated into an internal format before being translated into another native format. Examiner views the native formats as device specific and the internal format as device agnostic." Appellants respectfully disagree with the interpretation of an internal format as device agnostic, as will become more evident.

Whipple states at 0016 lines 10-14, "translating one or more API formats used by clients 18 to a format appropriate for the hub API, each such format preferably having a corresponding API adapter 24." Whipple further states, in part, in the abstract "The parameters (84) have one of multiple acceptable native formats. The request broker (50) determines the native format of the parameters (84) and communicates the parameters (84) in the native format to a selected one of multiple translators (24) for translation to an internal format, where each translator (24) is associated with a different native format."

Accordingly, Appellants understand Whipple to teach translating a request in a first native format into a format that is internal to the hub API (also referred to as an "internal format") and then translating the request that is in the hub API's internal format into a second native format (see Whipple 0016 lines 10-14, abstract, quoted herein).

Appellants respectfully submit that a format that is internal to a hub API is internal because it is only known by that hub API. Appellants respectfully submit that a format

that is only known by a hub API is not “agnostic.” Further, Appellants respectfully submit that a format that is only known by a hub API is the opposite of “agnostic” and, therefore, teaches away from the clear meaning of “agnostic.”

Further, Appellants respectfully submit that Whipple teaches away from “parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML).” For example, Whipple states at 0006 lines 9-12, “Certain embodiments of the invention may allow disparate remote clients to interact with a hub system using disparate corresponding data representations, such as Extensible Markup Language (XML), Electronic data Exchange (EDI), relational serialized object (e.g., JAVA), or relational formats, using a generic cross-firewall API mechanism.” Accordingly, Appellants understand Whipple to teach that XML is one of the disparate native formats that require translation into Whipple’s internal format. Appellants respectfully submit that since Whipple regards XML as one of Whipple’s native formats that requires translation into Whipple’s internal format, Whipple cannot be used to teach that XML can be used to represent “received device-agnostic policy implementation,” (emphasis added) as recited by Claim 1.

Further, Appellants respectfully submit since Whipple regards XML as one of Whipple’s native formats that requires translation into Whipple’s internal, Whipple teaches away from “received device-agnostic policy implementation,” (emphasis added) as recited by Claim 1.

Examiner’s Answer page 9 lines 5-18 for item B

Appellants respectfully submit that Corbin does not remedy the deficiencies in Whipple. For example, as discussed herein, Appellants do not understand either Whipple or Corbin to teach or suggest “a plurality of device-agnostic policy implementations... wherein a type of network device associated with a received device-agnostic policy implantation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML)...a plurality of device translators...each of said plurality of device translators

translating said device-agnostic policy implementation into corresponding device-specific implementations,” as recited by independent Claim 1.

For example, Corbin title is “Method and System for Representing a High-Level Programming Language Data Structure In A Mark-Up Language.” Corbin states at Col. 1 lines 15-28,

There are currently a myriad of high-level programming languages available for a programmer to use. Some of the more popular ones include Java, Java Script and C++. Each high-level language has its own way of defining data structures, such as arrays, integers, strings, and the like. However, a data structure written in the source code of one language generally cannot be compiled by a compiler of a different language. Thus, if a programmer is working on a system in which two or more different programming languages are being used, but all require access to the same data structure, then he or she is forced to write the data structure in each applicable language. This is particular cumbersome when the data structure needs to be changed during the debugging process.

Corbin also states at Col. 4 lines 13-16, “For example, a data structure may be written once in XML and translated into several high-level programming languages, such as C++, Java, ADA, Visual Basic, and Pascal.”

Accordingly, Appellants understand Corbin to teach using tags to create a data structure that is generic with respect to different programming languages (see Corbin title, Col. 1 lines 15-28, Col. 4 lines 13-16 quoted herein). Appellants respectfully submit that a data structure that is generic with respect to different programming languages does not remedy the deficiencies in Whipple.

Second, Appellants respectfully submit that since Whipple teaches away from “device-agnostic policy implementations,” “translating said device-agnostic policy implementation into corresponding device-specific implementations,” and “each device translator corresponding to a respective one of said plurality of network devices...each of said plurality of device translators translating said device-agnostic policy

implementation into corresponding device-specific implementations,” as discussed herein, there is no motivation to combine Corbin with Whipple.

policy implementations

Appellants reiterate that Appellants do not understand either Whipple or Corbin to teach or suggest “policy implementations.” For example, Appellants understand Whipple to teach translating requests (see Whipple’s abstract) and Corbin to teach a data structure (see Corbin Col. 4 lines 13-16). Appellants do not understand either a request or a data structure to teach or suggest “policy implementations.”

CONCLUSION

In view of the above remarks, Appellants continue to assert that Claims 1-4 and 6-18 are patentable over Whipple and Corbin, for reasons presented above and for reasons previously presented in the Appeal Brief.

Respectfully submitted,

WAGNER BLECHER LLP

Dated: 02/21/2012

/John P. Wagner, Jr./

\_\_\_\_\_  
John P. Wagner, Jr.

Registration Number:       35,398

WAGNER BLECHER LLP  
123 Westridge Drive  
Watsonville, CA 95076  
(408) 377-0500

82098653

Application No.: 10/690,605  
Group Art Unit: 2436